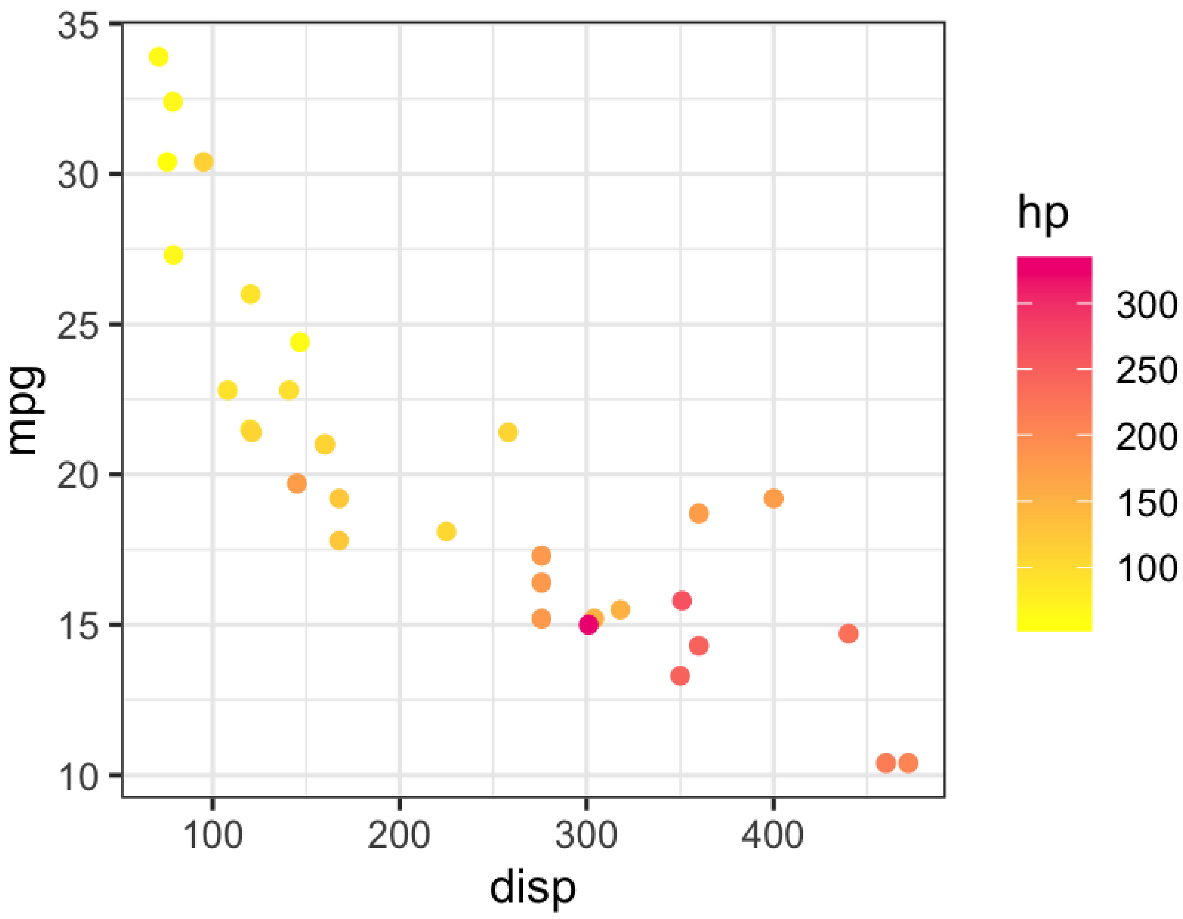
Scatterplots

Color Gradient

```
ggplot(data = mtcars,
mapping = aes(
x = disp,
y = mpg,
color = hp
)) + geom_point() +
scale_color_gradient(low="#FFFF00", high="#F00080") +
labs(x="Displacement", y = "Fuel efficiency", color="Horsepower"
```

- We are using a color gradient to signify low and high values



Simple Scatterplot

ggplot(data = mtcars,

mapping = aes(x = disp,

y = mpg)) +

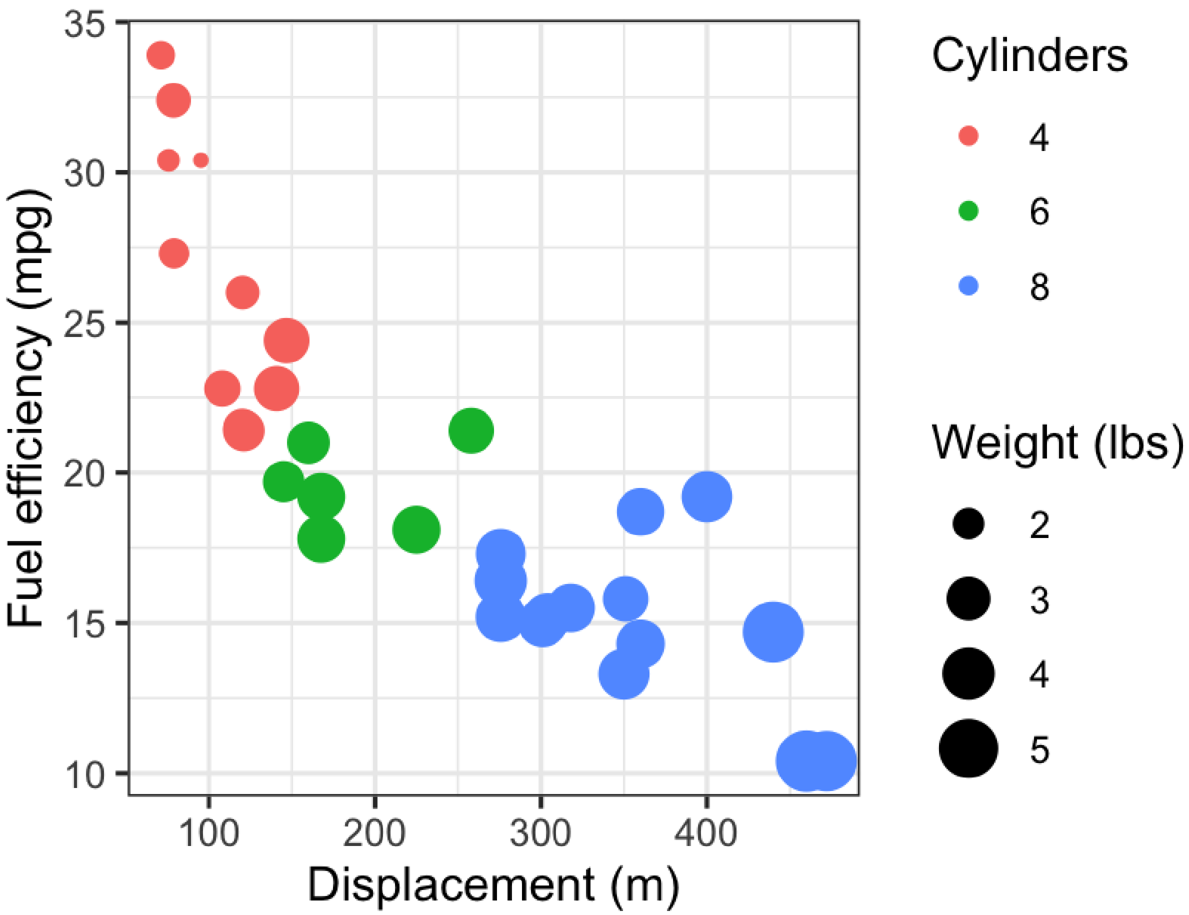
geom_point() +

labs(x="Displacement (m)", y = "Horsepower (mpg)")

Scatterplot with diff color for each group

```
ggplot(data = mtcars,
mapping = aes(x = disp,
y = mpg,
color = as.factor(cyl),
size = wt)) +
geom_point() + labs(x="Displacement (m)",
y = "Fuel efficiency (mpg)",
color="Cylinders",
size="Weight (lbs)")
```

*as.factor for numeric variables, remove for categorical variables

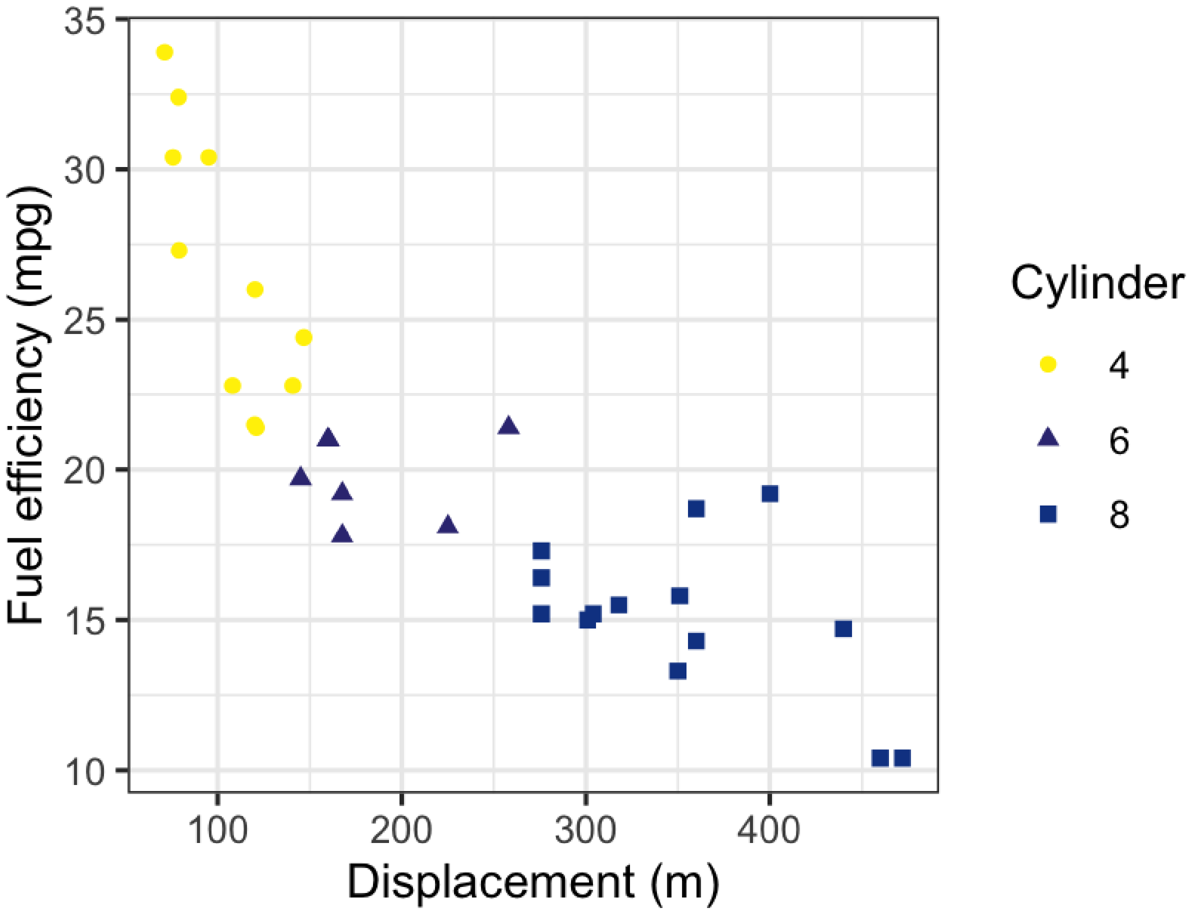


Adding Color and a Manual Color Scale

ggplot(data = mtcars,

```
mapping = aes(x = disp,
y = mpg,
color =
as.factor(cyl),
shape =
as.factor(cyl))) +
geom_point() +
scale_color_manual(values = c("
#E69F00",
"
#56B4E9",
"
#009E73")) + labs(x=”Displacement”,
 y = “Fuel efficiency (mpg)”,
 color=”Cylinder”,
shape=”Cylinder”)
```

*We are defining a color scale for the points (specifying which colors to use)



Adding Shape and A Manual Shape Scale

ggplot(data = mtcars,

mapping = aes(x = disp,

y = mpg,

color =

as.factor(cyl),

shape =

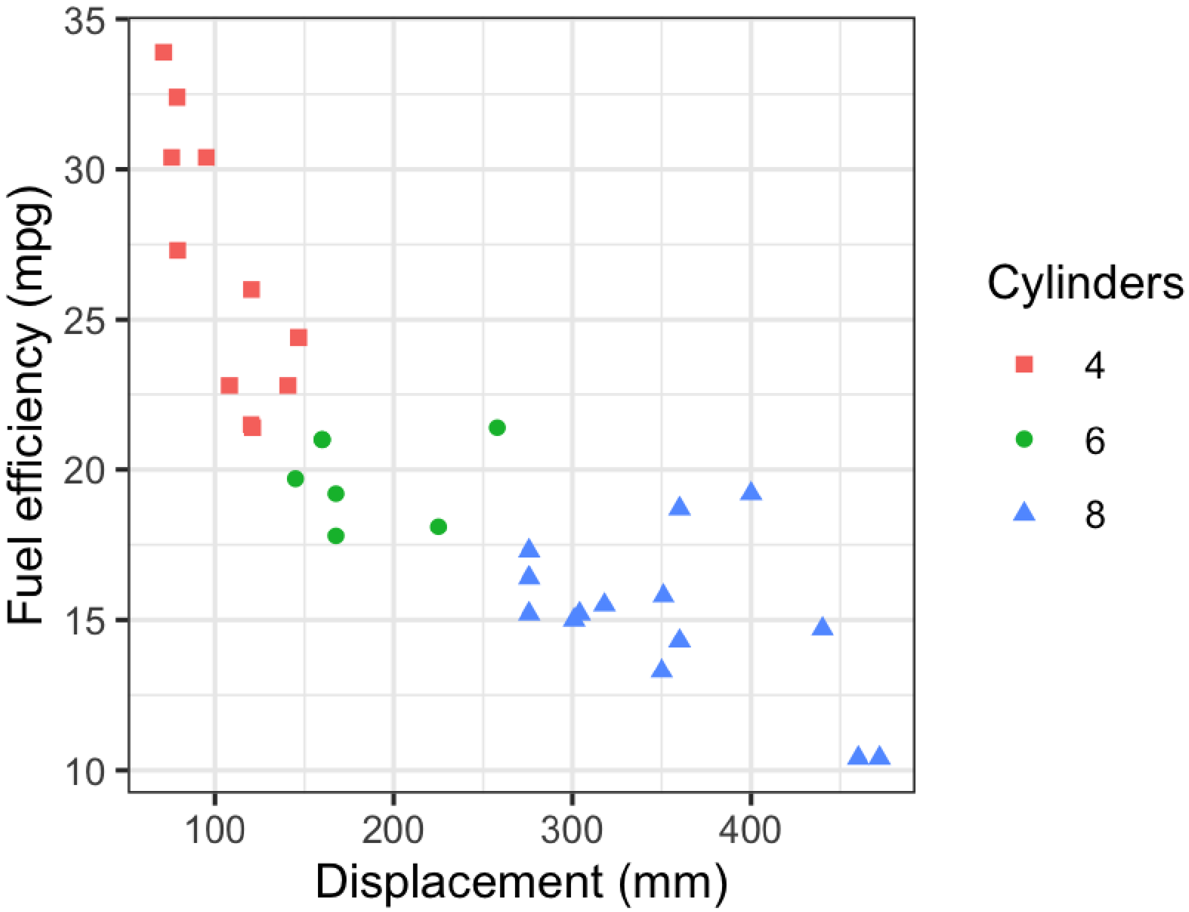
as.factor(cyl))) +

geom_point() +

scale_shape_manual(values = c(

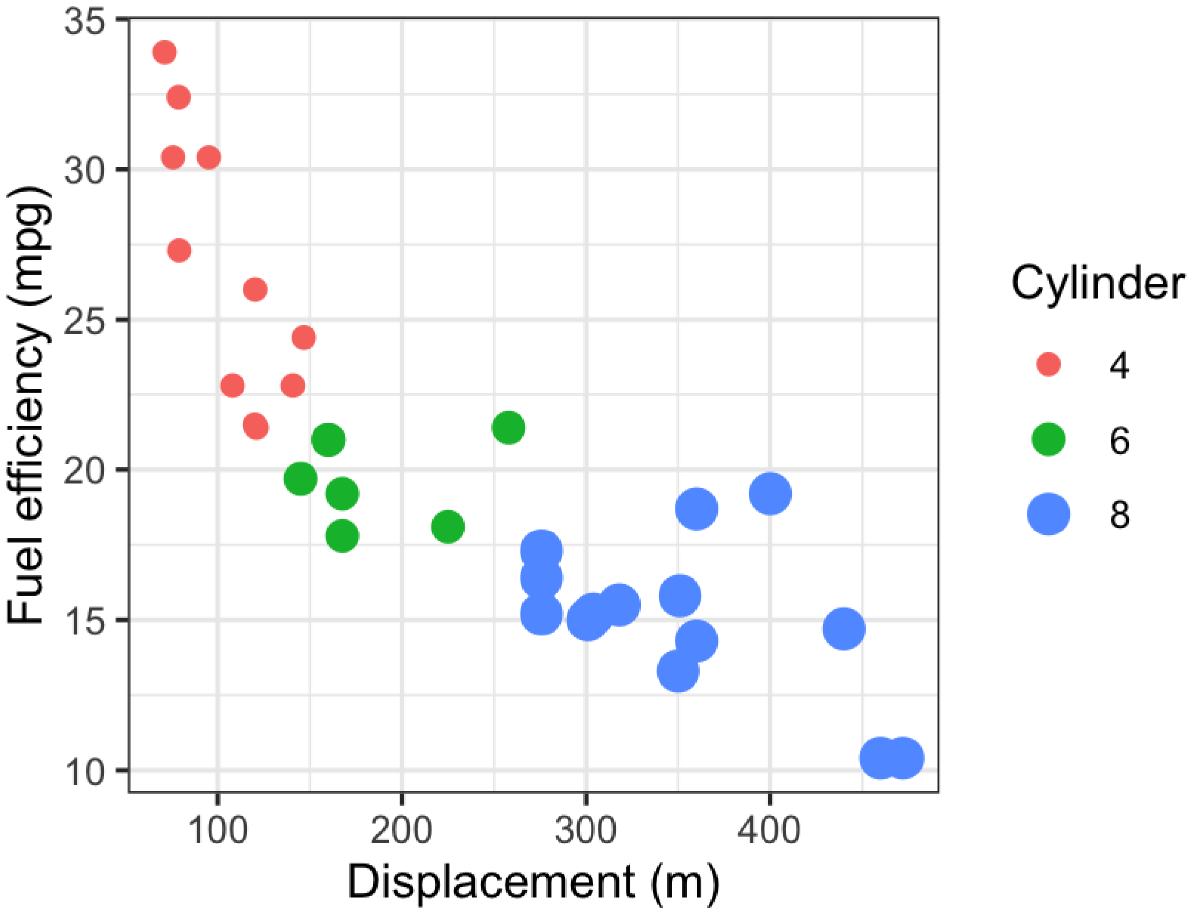
23,24,25))

*We are defining what shapes to use for the points



Adding Size and a Manual Size Scale (Discrete)

```
ggplot(data = mtcars,
mapping = aes(x = disp,
y = mpg,
color =
as.factor(cyl),
size =
as.factor(cyl))) +
geom_point() +
scale_size_manual(values = c(2,3,4)) +
labs(x="Displacement (m)", y="Fuel efficiency (mpg)", color="Cylinder",
size="Cylinder")
```



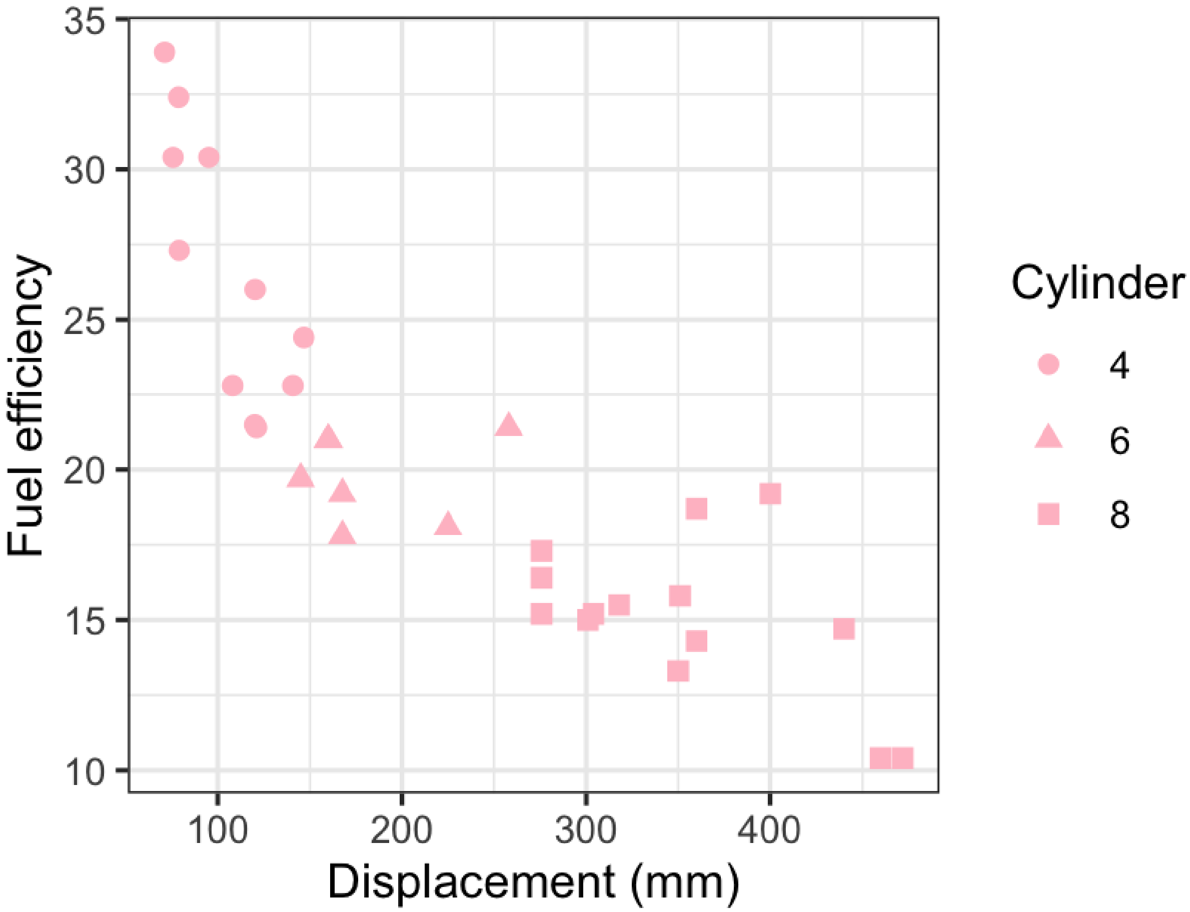
Adding Continuous Size and Size Scale

```
ggplot(data = mtcars,
mapping = aes(
x = disp,
y = mpg,
size = wt,
shape = as.factor(cyl)
)) + geom_point() + scale_size_continuous(range = c(0.5,5))
```

Independent changes from Data

```
ggplot(data = mtcars,
mapping = aes(x = disp,
y = mpg,
shape =
as.factor(cyl)))+
geom_point(color = "black", size=2) + labs(x="Displacement (mm)", y="Fuel
efficiency",
shape="Cylinder")
```

*Each point will have a pink color and a size equals 2



Complex Scatterplots

```
ggplot(data = mtcars,
    mapping = aes(x = disp,
    y = mpg,
    size = wt,
    shape = as.factor(cyl))) +
    geom_point() + theme_minimal() +
    theme(legend.position="top") +
     xlab("Displacement(m)") +
    ylab("Horsepower (mpg)") +
    guides(size = guide_legend(title="weight (1000lbs)", order = 1),
    shape = guide_legend(title="Cylinders", order = 2)) +
    theme_minimal() +
    theme(legend.position="top")
```

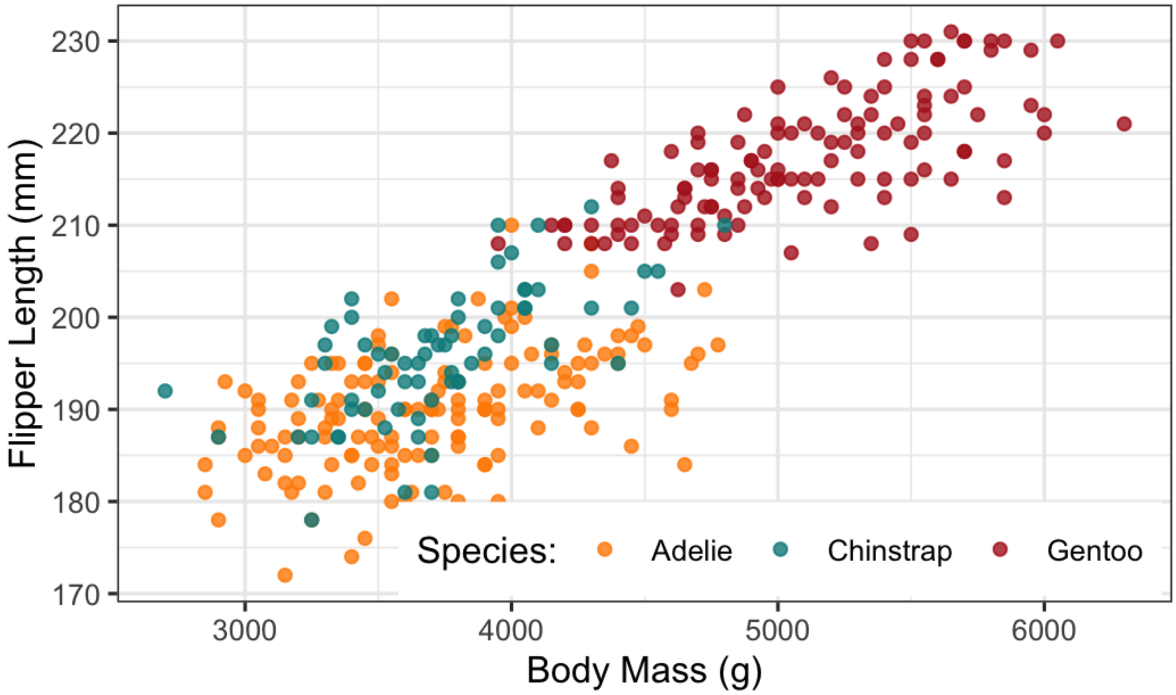
Fixing The Legend

```
ggplot(data = penguins_complete,
mapping = aes(
    x = body_mass_g,
    y = flipper_length_mm,
    color = as.factor(species)
)) + geom_point() +
xlab("Body mass (g)") +
ylab("Flipper length (mm)")+
labs(color="Species")+ theme_minimal()
```

Specific Scatterplot Example

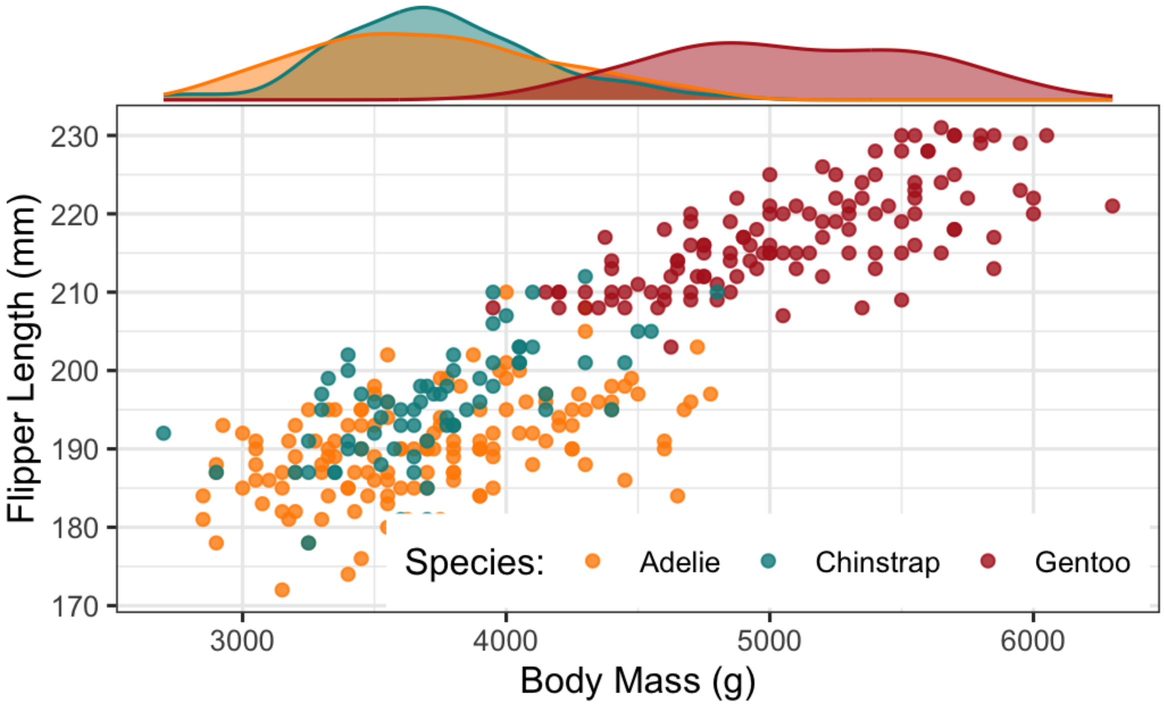
library(palmerpenguins)
penguins_complete ← na.omit(penguins)

~~~
```{r ScatterPlot}

penguin_scatter <- ggplot(penguins_complete,
                  aes(x = body_mass_g,
                  y = flipper_length_mm,
                  colour = species)) +
  geom_point(alpha = 0.8) +
  scale_colour_manual(values = c("Adelie" = "darkorange",
                                 "Chinstrap" = "darkcyan",
                                 "Gentoo" = "firebrick")) +
  labs(x = "Body Mass (g)",
       y = "Flipper Length (mm)",
       colour = "Species:") +
   theme(legend.direction="horizontal", legend.justification = c(1, 0),
        legend.position = c(0.99, 0.01))

penguin_scatter
~~~



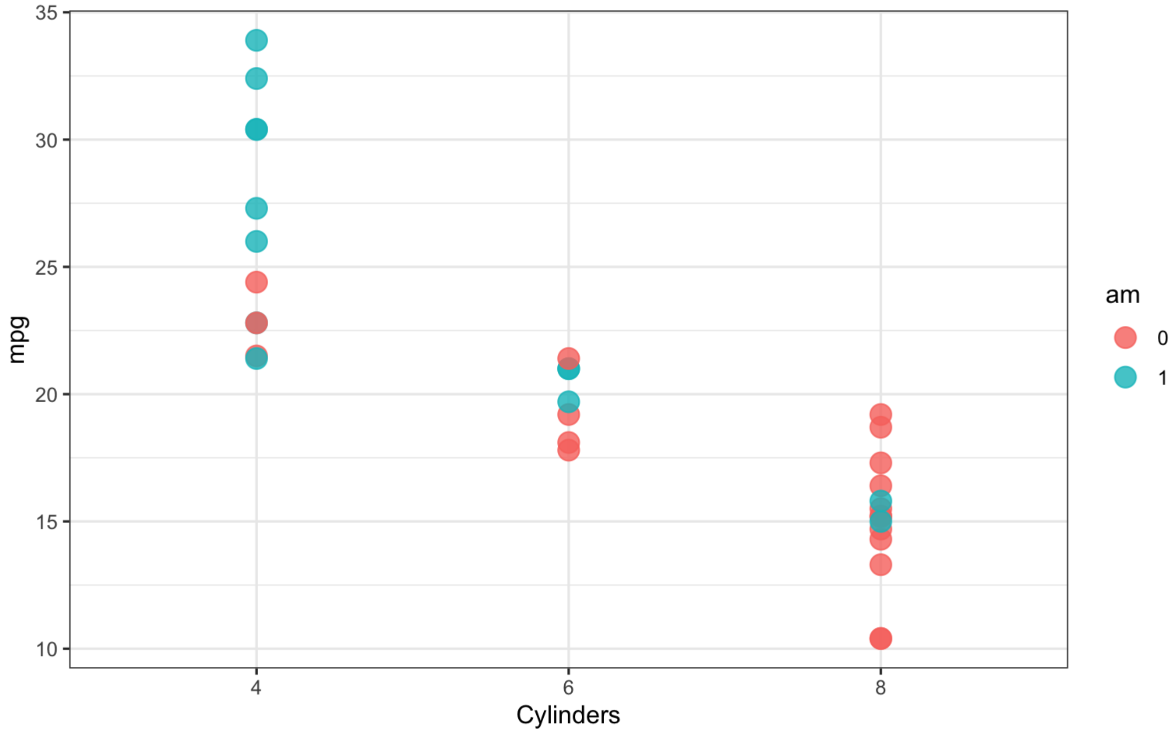
```
ggMarginal(penguin_scatter,
                    type = "density",
                     margins = "x",
          groupColour = TRUE,
             groupFill = TRUE)
```

*Adding a marginal graph on top of an existing scatterplot



Challenge

```
ggplot(data = mtcars,
       mapping = aes(
           x = as.factor(cyl),
           y = mpg,
           color = as.factor(am))) + geom_point(size=4, alpha
guides(color = guide_legend(title="am")) +
  labs(x ="Cylinders", y = "mpg")
  # discrete
```


35
30
25
20
15
10
mpg
4
6
8
Cylinders
am
0
1